

WHAT IS CLAIMED IS:

- 1 1. A method for managing a link stack comprising the steps of:
2 for a first type instruction:
3 setting a first data value corresponding to a first address stored in said
4 link stack in a first portion of an entry in a queue having a plurality of entries; and
5 for a second type instruction:
6 setting a current value of a first pointer into said link stack in a first
7 register; and
8 setting said current value of said first pointer in a second portion of
9 said entry in said queue.
- 1 2. The method of claim 1 further comprising the steps of:
2 for said second type instruction, reading from said link stack a second address
3 stored at a stack entry at said current value of said first pointer; and
4 storing said second address in a second register.
- 1 3. The method of claim 1 wherein said first type instruction is a "push" type
2 instruction and said second type instruction is a "pop" type instruction.
- 1 4. The method of claim 1 further comprising the step of, for a first type
2 operation, setting said first data value in a third register.

1 5. The method of claim 4 further comprising the step of, for third type
2 instruction, setting a second data value from said second register in a said first portion
3 of said entry.

1 6. The method of claim 1 wherein said steps recited therein are performed in
2 response to a fetch of a corresponding one of said first type instruction and said
3 second type instruction

1 7. The method of claim 2 further comprising the steps of:
2 receiving a pointer value in response to a pipeline flush, said pointer value
3 operable for pointing into said queue; and
4 setting a current pointer value of said first pointer to a value in said second
5 portion of an entry pointed to by said pointer value.

1 8. The method of claim 7 further comprising the steps of:
2 retrieving a value from said second register; and
3 comparing said value from said retrieving step with a value from said second
4 portion of said entry pointed to by said pointer value.

1 9. The method of claim 8 further comprising the steps of:
2 in response to a compare in said comparing step, comparing said current value
3 of said first pointer with a value in said first register; and
4 in response to a compare of said current value of said first pointer and said
5 value in said first register, setting said value in said second register in said link stack
6 at a location pointed to by said current value of said first pointer decremented by one.

- 1 10. A method of managing a link stack comprising:
2 for a first type instruction:
3 setting, in a first register, a first data value from a link stack entry
4 pointed to by a current value of a first pointer;
5 setting said current value of said first pointer in a second register; and
6 setting a second data value in said link stack at an entry pointed to by
7 said current value of said first pointer; and
8 for a second type instruction:
9 setting said decremented value of said first pointer into a second
10 portion of an entry in a queue having a plurality of entries.
- 1 11. The method of claim 10 further comprising the steps of:
2 for said first type operation:
3 setting a first Boolean value in a third register;
4 setting a current value of said second pointer in a fourth register, said
5 second pointer operable for pointing into a queue for tracking instructions; and
6 for said second type operation:
7 decrementing a current value of said first pointer by one;
8 reading from said link stack a first address stored at a stack entry at
9 said decremented current value of said first pointer; and
10 incrementing a current value of a second pointer, said second
11 pointer pointing to an entry point in said queue.

1 12. The method of claim 11 further comprising the steps of, in response to a
2 completion of an instruction:
3 comparing a current value of a third pointer and a value in said fourth register,
4 wherein said fourth pointer is operable for deallocating an entry in said queue.

1 13. The method of claim 12 further comprising, in response to a compare in said
2 comparing step, the steps of:
3 placing a second Boolean value in said fourth register; and
4 decrementing said fourth pointer.

1 14. The method of claim 11 further comprising the steps of:
2 determining if a value in said fourth register is between a current value of said
3 second pointer and a pointer value received in response to a pipeline flush; and
4 determining if a current value in said third register corresponds to said first
5 Boolean value.

1 15. The method of claim 14 further comprising the step of setting a current value
2 from said first register in a link stack entry pointed to by a value from said second
3 register in response to said determining steps evaluating logically true.

1 16. The method of claim 13, wherein said first Boolean value represents logical
2 true and said second Boolean value represents logical false.

1 17. The method of claim 14 further comprising the step of setting a current value
2 of said second pointer to said pointer value received in response to said pipeline flush.

1 18. A data processing system comprising:
2 a central processing unit (CPU), said CPU including:
3 a link stack; and
4 first logic operable for, for a first type instruction, setting a first data
5 value corresponding to a first address stored in said link stack in a first portion of an
6 entry in a queue having a plurality of entries, and for a second type instruction, setting
7 a current value of a first pointer into said link stack in a first register, and setting said
8 current value of said first pointer in a second portion of said entry in said queue

1 19. The system of claim 18 wherein said CPU further comprises, second logic
2 operable for, for said second type instruction, reading from said link stack a second
3 address stored at a stack entry at said current value of said first pointer, and storing
said second address in a second register.

1 20. The system of claim 18 wherein said first type instruction is a "push" type
2 instruction and said second type instruction is a "pop" type instruction.

1 21. The system of claim 18 wherein said CPU further comprises, third logic
2 operable for, for a first type operation, setting said first data value in a third register.

1 22. The system of claim 21 wherein said CPU further comprises fourth logic
2 operable for, for third type instruction, setting a second data value from said second
3 register in a said first portion of said entry.

1 23. The system of claim 18 wherein said first logic sets said first data value, for
2 said first type instruction, and sets said current value of said pointer, for said second
3 type instruction in response to a fetch of a corresponding one of said first type
4 instruction and said second type instruction.

1 24. The system of claim 19 wherein said CPU further comprises:
2 fifth logic operable for receiving a pointer value in response to a pipeline
3 flush, said pointer value operable for pointing into said queue; and
4 sixth logic operable for setting a current pointer value of said first pointer to a
5 value in said second portion of an entry pointed to by said pointer value.

1 25. The system of claim 24 wherein said CPU further comprises:
2 seventh logic operable for retrieving a value from said second register; and
3 eighth logic operable for comparing said value from said retrieving step with a
4 value from said second portion of said entry pointed to by said pointer value.

1 26. The system of claim 25 wherein said CPU further comprises:
2 ninth logic operable for, in response to a compare in said comparing step,
3 comparing said current value of said first pointer with a value in said first register;
4 and
5 tenth logic operable for, in response to a compare of said current value of said
6 first pointer and said value in said first register, setting said value in said second
7 register in said link stack at a location pointed to by said current value of said first
8 pointer decremented by one.

11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533

1 28. A data processing system comprising:
2 a central processing unit (CPU), said CPU including:
3 a link stack;
4 a first register;
5 a second register
6 first logic operable for, for a first type instruction, setting, in said first
7 register, a first data value from a link stack entry pointed to by a current value of a
8 first pointer, setting said current value of said first pointer in said second register, and
9 setting a second data value in said link stack at an entry pointed to by said current
10 value of said first pointer; and
11 second logic operable for, for a second type instruction,
12 decrementing a current value of said first pointer, reading from said link stack a first
13 address stored at a stack entry at said decremented current value of said first pointer,
14 setting said decremented value of said first pointer into a second
15 portion of an entry in a queue having a plurality of entries, said queue operable for
16 tracking instructions and incrementing a current value of a second pointer, said
17 second pointer pointing to an entry point in said queue.

1 29. The system of claim 28 wherein said CPU further comprises:
2 a third register;
3 a fourth register;
4 third logic operable for, for said first type operation, setting a first
5 Boolean value in said third register, and setting a current value of a second pointer in
6 said fourth register, said second pointer operable for pointing into said queue; and
7 wherein said second logic is further operable for, for said second type instruction.

1 30. The system of claim 29 wherein said CPU further comprises:

2 fourth logic operable for, in response to a completion of an instruction,
3 comparing a current value of a third pointer and a value in said fourth register,
4 wherein said fourth pointer is operable for deallocating an entry in said queue.

1 31. The system of claim 30 wherein said CPU further comprises, fifth logic
2 operable for, in response to a compare in said fourth logic, placing a second Boolean
3 value in said fourth register, and decrementing said fourth pointer.

1 32. The system of claim 29 wherein said CPU further comprises:
2 sixth logic operable for determining if a value in said fourth register is
3 between a current value of said second pointer and a pointer value received in
4 response to a pipeline flush; and
5 seventh logic operable for determining if a current value in said third register
6 corresponds to said first Boolean value.

1 33. The system of claim 32 wherein said CPU further comprises eighth logic
2 operable for setting a current value from said first register in a link stack entry pointed
3 to by a value from said second register in response to said sixth and seventh logic
4 outputting a logical true value.

1 34. The system of claim 31 wherein said first Boolean value represents a logical
2 true value and said second Boolean value represents a logical false value.

1 35. The system of claim 32 wherein said CPU further comprises ninth logic

- 2 operable for setting a current value of said second pointer to said pointer value
- 3 received in response to said pipeline flush.

1 36. A method for managing a link stack comprising the steps of:
2 for a first type instruction:
3 setting a first address in a first portion of an entry in a queue having a
4 plurality of entries; and
5 setting a current value of a first pointer into said link stack in a second
6 portion of said entry in said queue; and.
7 for a second type instruction:
8 setting a second current value of said first pointer in said second
9 portion of said entry in said queue.

1 37. The method of claim 36 further comprising the steps of:
2 for said second type instruction, reading from said link stack a second address
3 stored at a stack entry at said second current value of said first pointer.

1 38. The method of claim 36 wherein said first type instruction is a "push" type
2 instruction and said second type instruction is a "pop" type instruction.

1 39. The method of claim 36 wherein said steps recited therein are performed in
2 response to a fetch of a corresponding one of said first type instruction and said
3 second type instruction.

1 40. The method of claim 36, for a third type instruction comprising the steps of:
2 receiving a pointer value from, said pointer value operable for pointing into
3 said queue; and
4 setting a current pointer value of said first pointer to a value in said

5

1

2

[illegible]

1 42. A data processing system comprising:
2 a central processing unit (CPU), said CPU including:
3 a link stack;
4 a queue having a plurality of entries, said queue operable for tracking
5 instructions; and
6 first logic operable for, for a first type instruction, setting a first
7 address into a stack entry in said link stack at a current value of a first pointer and
8 setting a current value of said first pointer into a first portion of an entry in said
9 queue.

1 43. The system of claim 42, wherein said CPU further comprises, second logic
2 operable for, for a second type instruction, setting a second current value of said first
3 pointer in said second portion of said entry in said queue.

1 44. The system of claim 43, wherein said second logic, for said second type
2 instruction, is further operable to read from said link stack a second address stored at
3 a stack entry at said second current value of said first pointer.

1 45. The system of claim 42, wherein said first type instruction is a "push" type
2 instruction.

1 46. The system of claim 43, wherein said second type instruction is a "pop" type
2 instruction.

1 47. The system of claim 42, wherein said CPU further comprises, third logic
2 operable for, for a third type operation, receiving a pointer value, said pointer value
3 operable for pointing into said queue and setting a current pointer value of said first
4 pointer to a value in said second portion of an entry pointed to by said pointer value.

1 48. The system of claim 47, wherein said third type instruction is a "flush" type
2 instruction.